Peters, A., Oikonomou, G., & Zervas, G. (2018). In compute/memory dynamic packet/circuit switch placement for optically disaggregated data centers. *IEEE/OSA Journal of Optical Communications and Networking*, *10*(7), B164-B178. https://doi.org/10.1364/JOCN.10.00B164

In Compute/Memory Dynamic Packet/Circuit Switch Placement for Optically Disaggregated Data Centres

Adaranijo Peters, George Oikonomou and Georgios Zervas

Abstract—Network function services on conventional hybrid data centres (DCs) architectures such as HELIOS are hard-wired and dedicated to specific network resources. This limits flexibility and performance to handle diverse traffic patterns. Furthermore, disaggregation of server resources has shown promising potential to improve resource utilization which has been a limitation of conventional server-centric DCs. This paper presents a reconfigurable hybrid disaggregated DC (dRedBox) architecture which combines the concept of server resource disaggregation with cutting edge software, electronic and optical technologies. The dRedBox architecture provides a remarkable amount of flexibility and connectivity through hardware based multi-layer network function service programmability. This allows for multi-layer network services to be dynamically deployed at run-time to network resources and in turn handle diverse traffic patterns. Furthermore, this study proposes algorithms and strategies for selecting and deploying electronic packet switching and optical circuit switching function services to implement Virtual Machine network requests across dRedBox and conventional hybrid disaggregated architectures under different traffic patterns. Finally, the performance of the various strategies on the dRedBox and conventional hybrid disaggregated DC architectures are evaluated in terms of blocking probability, energy efficiency, network utilization and cost. Extensive results show that at 10% blocking probability, dRedBox architecture achieves 100% gain on VM placement and 92% energy savings compared to conventional hybrid disaggregated architectures.

Index Terms—Data centres; Optical circuit switching; Electronic packet switching; Multiplexing and Network topologies.

I. Introduction

Global data centre (DC) traffic is projected to rise in the next few years with about 77% of the DC traffic remaining within the DC [1]. With conventional DCs experiencing drawbacks in server resource utilization, power consumption and scalability, novel solutions must be explored to effectively manage this traffic trend. Disaggregation of servers into separate standalone CPU, memory, storage and network resource pools has been projected in [2,3] as a promising solution to improve resource utilization, efficiency, modularity and upgradeability in DCs. In addition, hybrid DC architectures employing both electronic and optical technologies [4-6] and all-optical technologies [7-9] have been explored and have demonstrated benefits in terms of power consumption, modularity and cost in comparison to conventional DCs. Therefore, the combination of server resource disaggregation with hybrid technologies provides a promising solution to mitigate the drawback experienced by conventional server-centric DCs. However, for this concept to materialize, there are several challenges that need to be addressed. First, the authors in [2] stated that communication network will be a challenge for the realization of server resource disaggregation. This is because communication between resources (e.g. CPU to CPU and CPU to memory) for Virtual Machine (VM) placement which occurred within a server will now be distributed throughout the DC network. Secondly, existing hybrid DCs still suffer from one main drawback; the network function services in these hybrid DCs are hardwired with fixed electronic packet switching (EPS) to optical circuit switching (OCS) proportionalities and are dedicated to specific network resources. This limits the ability to dynamically deploy or upgrade network function services, which in turn limits the flexibility and performance of the DC to efficiently manage diverse network traffic requirements.

The dRedBox (Disaggregated Recursive Data Centre in a

Manuscript received January 29, 2018.

Adaranijo Peters and George Oikonomou are with Department of Electrical and Electronic Engineering, University of Bristol, Bristol, BS8 1UB, United Kingdom (email: adaranijo.peters@bristol.ac.uk).

Georgios Zervas is with Department of Electronic and Electrical Engineering, University College London, London, WC1E 7JE, United Kingdom.

Box) architecture offers promising solutions to overcome these challenges. This architecture combines server resource disaggregation with state of the art software, optical and electronic technologies to deliver a DC architecture with efficient resource utilization, low power consumption, low latency, high throughput and non-disruptive upgradeability [3]. The dRedBox architecture provides improved flexibility and connectivity compared to conventional hybrid DCs architectures due to deep multi-layer network function service programmability at end nodes (i.e. embedded CPU, memory) [10]. Unlike conventional hybrid DCs where a fixed amount of network resources for server input/output (I/O) ports are dedicated to either optical circuit or electronic packet network, the dRedBox architecture allows for dynamic and run-time deployment of on-chip packet/circuit switching service to any I/O port in order to handle variable network requirements. Thus, VM requests can be implemented using a custom-built network topology with a network service chain spanning over multiple network hops. The introduction of these amount of flexibility and programmability features opens up challenges on how to effectively deploy and manage these network function services to deliver optimum network performance when serving VMs.

Typically, a VM request specifies the amount of CPU cores, memory and bandwidth connectivity required. Furthermore, the deployment of a VM request in disaggregated DCs requires two main phases: IT resource and network resource allocation. Factors such as resource availability, resource capacity, the location of the different resource types across the DC network (i.e. how CPU and memory resources are arranged), the network fabrics and allocation strategies should be considered when deploying VMs. Several studies have performed simulations studies on IT and network resource allocation [11-13] for the deployment of VMs in disaggregated DCs. However, to the best of our knowledge, apart from our preliminary work in [14], no other study has proposed algorithms or performed a simulation study to address the challenges associated with the deployment and allocation of EPS and OCS network function services to build VMs on disaggregated resources on various hybrid DC architectures. The main focus this paper is to investigate networking strategies and algorithms on how to select and allocate EPS/OCS services and resources in the most cost/power efficient way to serve VMs across different hybrid disaggregated DC architectures. Hence, it is assumed that IT resources requirements, i.e., CPU and memory, for a VM request have already been allocated and the requests presented in this study are network requests to build VMs on disaggregated resources (and not associated with the network bandwidth requests between VMs or a VM and the cloud or user), we define this type of request as a VM network request. The preliminary work in [14] reported an algorithm for selecting and deploying EPS/OCS services or a combination of both to generate custom-built network topologies to implement the VM network requests consisting of 2 CPU to 3 memory bricks. Furthermore, a performance analysis compared several simulation scenarios which include: the dRedBox and pure optical disaggregated rack architecture, dRedBox rack with heterogeneous tray architecture and dRedBox rack with homogenous tray architecture, and finally, dRedBox and classical architecture with heterogeneous tray architectures were discussed. This work is extended with the following contributions.

- The EPS/OCS placement and topology creation algorithm is extended to handle VM network requests consisting of any number of CPU to memory combinations and a cluster of racks for various DC architectures.

- Several networking strategies and algorithms for deploying and selecting network end points or electronic packet switches to perform statistical multiplexing when building EPS virtual/logical topologies are proposed.

- A comprehensive performance evaluation of the different networking strategies across clusters of dRedBox and conventional hybrid EPS/OCS disaggregated DC architectures is conducted and analyzed under different traffic patterns.

- A benchmark of the dRedBox architecture against conventional hybrid EPS/OCS disaggregated architectures in terms of blocking probability, network utilization, cost and energy efficiency is conducted.

The rest of this paper is organized as follows: Section II presents related work. Section III presents features of various hybrid disaggregated DC architectures. Section IV presents the implementation of VM network requests on the various hybrid disaggregated DCs. Section V presents the structure and working flow of the proposed algorithms. Section VI reports on simulation results and discussions. Finally, section VII concludes the paper.

II. Related Work

S. Han et. al. [2] discussed the potential benefits and challenges of disaggregation of server resources in DCs. It was reported that resource disaggregation in DCs offers potential benefits such as efficient resource utilization, modularity and upgradeability. However, the authors pointed out the DC communication network design as a key challenge which must be explored to reap the benefits of resource disaggregation. This is because communication between resources (e.g. CPU to memory) which has traditionally been restricted within a server, will now be distributed throughout the DC network. Recent works have implemented simulations studies on the performance of disaggregated DCs [11-13]. A. Pages et. al. [11] presented an integer linear programming and heuristic based model to optimize and evaluate the utilization of a disaggregated DC and compared it with server-centric DCs with the same resource pools. The study showed that disaggregated DCs offer better resource utilization than conventional server centric DCs. Also in [12], results from a mixed-integer linear programming and heuristic model showed that disaggregated DCs offer better power savings than server-

centric DCs for different types of VMs. Furthermore, the authors in [13], proposed and evaluated several algorithms for dynamic IT resource and network allocation for the dRedBox disaggregated DC. It is worth mentioning that disaggregated DC have been experimentally demonstrated in [15-19]. In particular, P. Gao et al [15] investigated network bandwidth and network latency performance requirements for DC disaggregation to avert application performance degradation. The study showed that for certain applications, 20-40 Gb/s for remote memory access can achieve minimal (under 10%) application performance degradation. In addition, the authors in [16] experimentally demonstrated communication between Multi-Processor System on Chip (MPSoC) hardware with multiple cores and remote access memory over 10 Gb/s lanes. The authors reported that a bandwidth of 582 Mib/s (~ 5 Gb/s) can be realized from a single CPU core to remote access memory. Also, the authors in [17] experimentally demonstrated remote memory access with 10 Gb/s link and up to 68% sustained memory bandwidth.

In recent years there has been a shift in focus from conventional electronic-based DCs to hybrid DCs incorporated with both electronic and optical technologies [4-6] or all-optical technologies [7-9]. This is because of the significant benefits that optical technology offers which includes: low latency, energy efficiency, scalability and high data rates. A detailed review of optical technologies for DCs was reported in [20]. In electronic/optical hybrid DCs architectures [4-6], electronic switching and optical switching fabrics are incorporated to perform EPS and OCS respectively. While in all-optical hybrid DCs, only all-optical technologies are incorporated. The switching fabric can be a combination of optical packet switching and OCS [7,8] or fast and slow optical switching [9]. Despite the benefits these hybrid DCs have demonstrated, the network function services are hardwired to specific network and server ports and cannot be re-configured. This drawback limits the optimization of service offering at the planning and dimensioning phase and in turn the reaction to unpredictable future demands which can lead to a decline in performance. This limitation has created a need to introduce dynamic network function service programmability in DCs. Network function service re-configurability on an FPGA platform has been proposed and demonstrated in [10]. The authors demonstrated network function service reconfiguration at run-time between layer 1 circuit switching service and layer 2 Ethernet packet switch service without service disruption or packet loss. In addition, a programmable hybrid DC with network function service programmability was presented and demonstrated in [21]. This work involved replacing static NICs on server resources with FPGA based interfaces and deploying network service chains to deliver multi-layer networking services.

III. Features of Hybrid Disaggregated Data Centre Architectures

A cluster of dRedBox DC architecture is displayed in Fig. 1(a). Each tray is composed of CPU bricks which hosts a multi-processor system on chip (MPSOC) that embeds multiple cores, and memory bricks. Both the CPU and memory bricks are embedded with a hybrid and

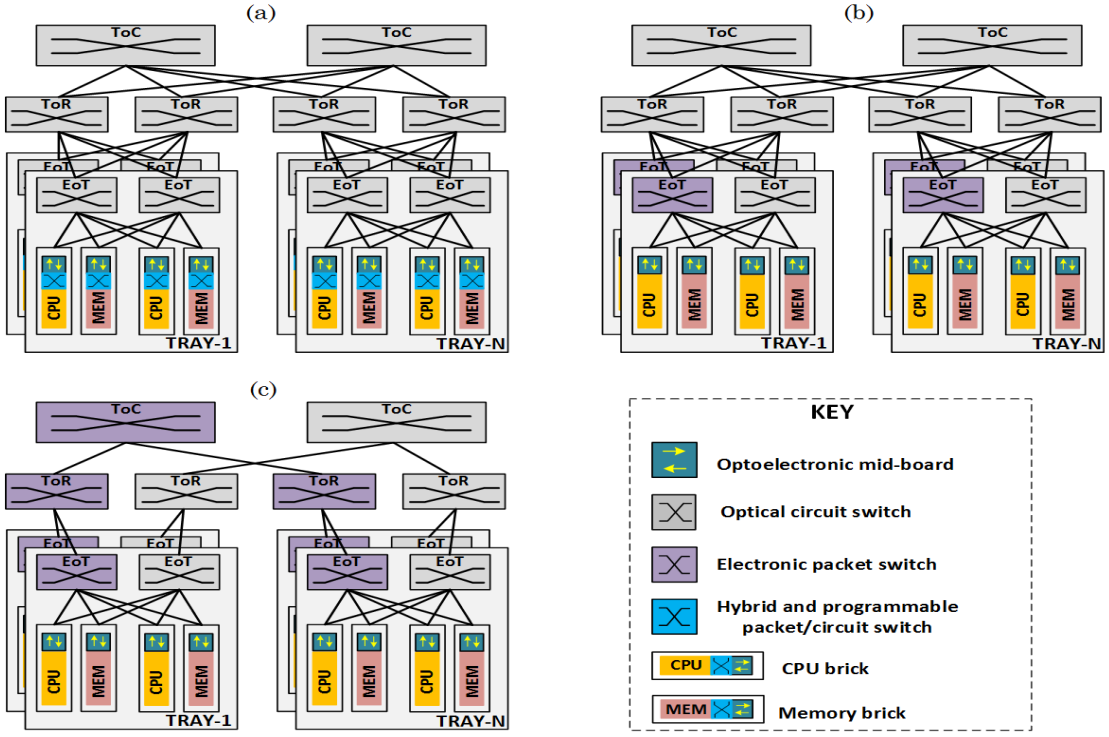

Fig. 1. Hybrid Disaggregated DC architectures: (a) dRedBox, (b) 1-Tier-H, (c) 3-Tier-H.

programmable electronic packet/circuit switch and optoelectronic mid-board with optical transceivers. The hybrid and programmable electronic packet/circuit switch can dynamically deploy electronic packet or circuit switching services to any I/O ports to handle different networking requirements. This allows for programmable ratio of packet to circuit switched services at runtime. The edge of tray (EoT), top of rack (ToR) and top of cluster (ToC) are a hierarchical layer of optical circuit switches which provide tray, rack and cluster level optical networking, respectively. Figure 1(b) and 1(c) presents conventional hybrid EPS/OCS architectures that have a fixed ratio of packet to circuit services. Figure 1(b) presents a cluster of 1-Tier hybrid disaggregated DC (1-Tier-H) architecture and Figure 1(c) presents a cluster of 3-Tier hybrid disaggregated DC (3-Tier-H) architecture. The 1-Tier-H tray is composed of CPU bricks which hosts MPSOC that embed multiple CPU cores, and memory bricks. Both CPU and memory bricks are embedded with optoelectronic mid-boards with optical transceivers and do not have any programmable electronic packet/circuit switch. EPS is carried out on dedicated EoT electronic packet switches and the network outside tray is supported by an optical network. The 3-Tier-H tray is similar to the 1-Tier-H tray. However, EPS is supported by dedicated ports on all network tiers (EoT, ToR and ToC). The remaining ports are supported by an optical network with a hierarchical layer of optical circuit switches (EoT, ToR and ToC) that provides optical networking.

IV. Implementation of VM network requests in Hybrid Disaggregated Data Centres

To understand the working principle of the previously described DC architectures, some examples of building VM network requests on the disaggregated DC architectures are highlighted. As previously mentioned in the introduction, it is assumed that the IT resource requirements for VMs have already been allocated and the requests considered in this study are network requests to build VMs on disaggregated resources. Figure 2(a) and 2(b) presents two VM network request matrices that are to be deployed in the dRedBox and conventional tray architectures displayed in Fig. 1. The transceivers embedded on the bricks and electronic packet switches are 10G transceivers and all the links in the DC network are bidirectional and support a capacity of 10Gb/s in each direction. The rows and columns of the VM network request matrices in Fig. 2 represents the CPU bricks and memory bricks respectively. Each element in the matrix represents the required bandwidth for a single directional network link to be established between a CPU brick and memory brick (i.e. the total bandwidth requirement for a complete roundtrip communication between a CPU and memory brick is 2 × the bandwidth requirement of a single directional transaction). The VM network request 1 in Fig. 2(a) indicates that CPU brick 1 requires network links to memory brick 2 and memory brick 4 with bandwidth requirements of 3 Gb/s and 2 Gb/s respectively, and CPU brick 3 requires network links to memory brick 2 and memory brick 4 with bandwidth requirements of 3 Gb/s and 1 Gb/s respectively. The VM network requests can be served using an EPS logical/virtual topology, an OCS logical/virtual topology or a combination of them. Building an EPS logical/virtual topology and serving the VM network request using EPS over OCS services is highly desirable due to the statistical multiplexing features that EPS technology provides. This in turn leads to the best utilization of network and IT resources. However, care should be taken to limit its use due to latency overhead.

(a)

CPU BRICK \ MEM BRICK	1	2	3	4	5	6
1	0	3	0	2	0	0
2	0	0	0	0	0	0
3	0	3	0	1	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0
6	0	0	0	0	0	0

VM network request 1

(b)

CPU BRICK \ MEM BRICK	1	2	3	4	5	6
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	8	0	8	0	9
4	0	0	0	0	0	0
5	0	9	0	7	0	7
6	0	0	0	0	0	0

VM network request 2

Fig. 2. VM network request matrices.

Figure 3 displays EPS/OCS service allocation scenarios for deploying VM network request 1 and VM network request 2 on the dRedBox tray and the conventional tray architecture. It is assumed that VM network request 1 arrives before VM network request 2 and the resources attached to VM network request 1 are still in use when VM network request 2 arrives. For both scenarios, each brick on the dRedBox and conventional tray supports 4 I/O ports i.e. 4 transceivers. Figure 3(a) illustrates the implementation of VM network request 1. When the bandwidth requirements between CPU and memory bricks of a VM network request are low (for example between 1 to 5 Gb/s) and each of the CPU bricks require network links to the same set of memory bricks, the multiple required network links can share network resources through statistical multiplexing and an EPS virtual/logical topology can be built to implement the VM network request. Since VM network request 1 has low bandwidth requirements between CPU and memory bricks, and CPU bricks 1 and 3 require network links to memory bricks 2 and 4, an EPS virtual/logical topology is built and EPS over OCS services are selected to implement VM network request 1 on the dRedBox and conventional tray architecture. (Details on the strategies for selecting EPS/OCS services are discussed in section V). In the dRedBox tray, CPU brick 1 is configured as an electronic packet switch to perform statistical multiplexing of flows from CPU bricks 1 and 3 to memory bricks 2 and 4, while in the conventional tray architecture, the statistical multiplexing of flows from CPU bricks 1 and 3 to memory bricks 2 and 4 occurs in the EoT electronic packet switch.

Comparing the two tray architectures in scenario 1 (Fig. 3(a)), the dRedBox architecture (Fig.3 (a) left) uses 6 brick I/O ports attached to 6 transceivers and 12 optical switch ports to implement the VM network request whereas the conventional architecture (Fig. 3 (a) right) uses 4 brick I/O ports attached to 4 transceivers and 8 electronic switch ports attached to 4 transceivers (i.e. the conventional tray architecture uses a total of 8 transceivers). In terms of the number of transceivers, the dRedBox architecture offers transceiver savings when compared to the conventional architecture. In terms of the number of switch (network) ports and brick I/O ports, the dRedBox architecture uses

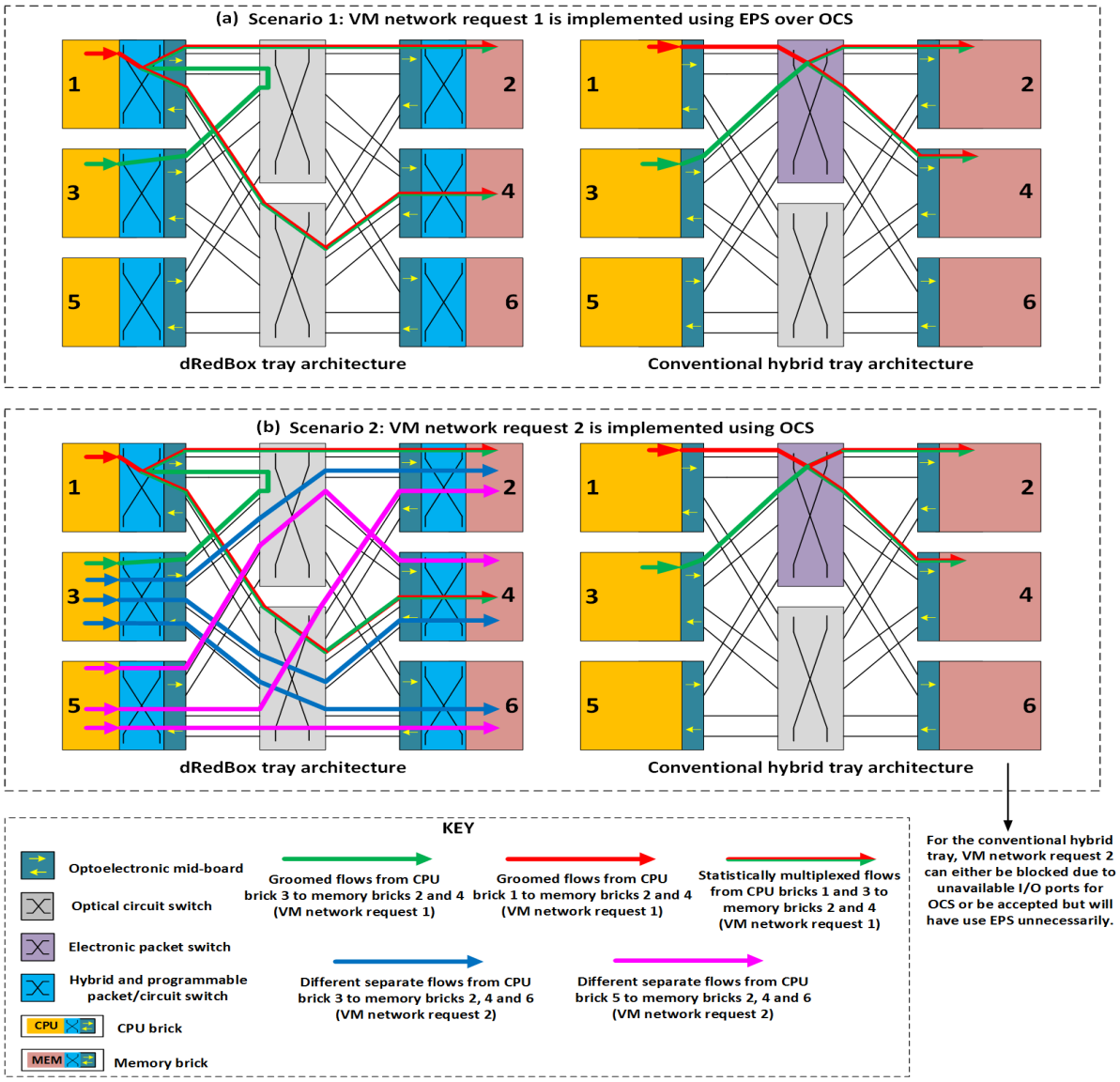

Fig. 3. Deployment of VM network requests using EPS/OCS services.

more switch ports and brick I/O ports than the conventional architecture. However, the dRedBox architecture is more cost-effective because network transmission is carried out over an optical switch unlike the conventional architecture which uses a power hungry electronic packet switch. Also, the dRedBox architecture also has the advantage of network function programmability and can deploy packet/circuit services to any of the I/O ports. Thus, either brick 1, 2, 3, or 4 can be configured to perform statistical multiplexing functions to merge flows from CPU bricks 1 and 3 to memory bricks 2 and 4.

Figure 3(b) illustrates the implementation of VM network request 2. The bandwidth requirements between CPU and memory bricks for VM network request 2 are high (between 6 to 10 Gb/s) and the multiple required network links cannot share network resources. Therefore, an OCS virtual/logical topology is created and OCS services are selected to implement VM network request 2. In the dRedBox architecture, a point to point link between each pair of CPU brick and memory brick is established. Thus, 3 I/O ports in CPU bricks 3 and 5, and 2 I/O ports in memory bricks 2, 4 and 6 are selected, and the network optical switches are configured to implement the VM network request. In the conventional tray architecture, VM network request 2 can either be blocked due to unavailable I/O ports on the CPU bricks to support OCS or be accepted but will have to use EPS resources unnecessarily.

V. Algorithm Structure

A simulator in Matlab consisting of several algorithms was developed to investigate the performance of networking strategies and network switching services to deploy VM network requests on the dRedBox, 1-Tier-H and 3-Tier-H architectures. The simulator builds custom multi-layer virtual/logical topologies and allocates network resources at

run-time to serve the VM network requests. The simulator is divided into three different stages as illustrated in Fig. 4. The operations of the three different stages are consolidated in algorithm 1. Table 1 displays the notations and description for algorithm 1.

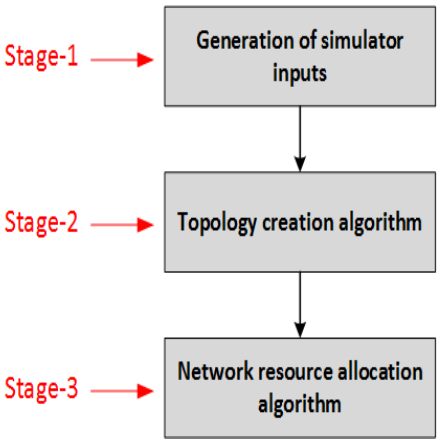

Fig. 4. Simplified framework for simulator.

A. *Simulator inputs*

The task achieved in the first stage of the algorithm is the generation of the simulator inputs. Line-1 of the pseudocode (Algorithm 1) presents the simulator inputs which include: $G(N,E)$, R and P. In particular, P is a set of subsets, where each subset P_i is a unique set or multiset of two elements which contains a possible way that different numbers of CPU and memory bricks in a VM network request can be paired to build EPS virtual/logical topologies. The first element $p_{i,1}$ and second element $p_{i,2}$ of each subset represents the number of CPU bricks and memory bricks respectively. An example of the set P is illustrated in Fig. 5. Assuming that the number of CPU bricks and memory bricks that can be allocated to a VM network request is between 2 to 4, P is generated with two steps. In the first step, the permutation with repetition of 2-element subsets from the set {2,3,4} is calculated. The set {2,3,4} is the range between the lowest number of CPU or memory bricks to the highest number of CPU or memory bricks that can be allocated in a VM network request which in this example is 2 and 4. In the second step, the generated 2-element subsets are sorted and rearranged starting from the subset with the elements that have the highest product to the subset with elements that have the lowest product. This is because building EPS virtual/logical topologies with higher number of CPU and memory brick combinations is beneficial, i.e., the higher the number of CPU and memory bricks that are paired to build EPS virtual/logical topologies, the more the number of network links in a VM network request can share network resources. This in turn leads to conservation of network resources for future use.

$$\{\{4,4\},\{4,3\},\{3,4\},\{3,3\},\{4,2\},\{2,4\},\{3,2\},\{2,3\},\{2,2\}\}$$

Fig. 5. Set P.

TABLE I

Symbol	Description
$G(N,E)$	Graph representing a transparent DC network topology where N is the set of network nodes (Bricks, EoTs, ToRs and ToCs) and E is the set of physical links.
R	Set of sets where each set R_i represents a VM network request to be deployed in the DC network.
$N_{R_i}^{CPU}$	Set of CPU bricks in R_i.
$N_{R_i}^{MEM}$	Set of memory bricks in R_i.
B	VM network request matrix where each element is a bandwidth requirement for a network link between a CPU brick in $N_{R_i}^{CPU}$ and a memory brick in $N_{R_i}^{MEM}$. The rows and columns of the VM network request matrix represents the CPU bricks and memory bricks respectively.
P	Set of subsets where each subset P_i is a unique set or multiset of two elements and represents a possible way that different numbers CPU and memory bricks in a VM network request can be paired to build EPS virtual/logical topologies. The first element $p_{i,1}$ and the second element $p_{i,2}$ of each subset represents the number of CPU bricks and memory bricks respectively.
C	Set of subsets where each subset C_i contains a possible combination of the CPU bricks in $N_{R_i}^{CPU}$, and each element $c_{i,j}$ represents the j^{th} CPU brick in C_i.
CM	Set of memory bricks where element represents a memory brick which requires a network link from all the CPU bricks in C_i.
M	Set of subsets where each subset M_i contains a possible combination of the memory bricks in CM, and each element $m_{i,j}$ represents the j^{th} memory brick in M_i.
V	A set which is the union all elements in C_i and M_i.
$TXcap$	Capacity of a transceiver.
$tb_{c_{i,j}}$	Sum of required bandwidth of network links from a CPU brick in C_i to all memory bricks in M_i.
$tb_{m_{i,j}}$	Sum of required bandwidth of network links from all CPU bricks in C_i to a memory brick in M_i.
$x_{c_{i,j}}$	Equals 1 if $tb_{c_{i,j}} \leq TXcap$, otherwise equals 0.
$x_{m_{i,j}}$	Equals 1 if $tb_{m_{i,j}} \leq TXcap$, otherwise equals 0.
K	Set which contains candidate bricks or electronic packet switches which can be used to build EPS virtual/logical topologies.
L	Number of candidate bricks or electronic packet switches in K.

TDB	Database for EPS virtual/logical topologies.
OT	Set which contains the remaining network links in B that have not been used to create an EPS virtual/logical topology.
Z	Set of sets where each subset Z_s represents a created topology for the VM network request R_i, and element $z_{s,d}$ represents the dth link in Z_s.
NZ	Number of generated topologies in Z.
f_{Z_s}	Number of links to established in Z_s.
$b_{z_{s,d}}$	Required bandwidth for link $z_{s,d}$.
$eb_{z_{s,d}}$	Bandwidth in an existing network path for link $z_{s,d}$.
$h_{z_{s,d}}$	Equals to 1 if resources are available for link $z_{s,d}$, otherwise equals 0.

B. Topology creation algorithm

The overall task achieved in the second stage of the algorithm is the creation of custom network topologies to implement the VM network request, the topologies could either be an EPS virtual/logical topology, an OCS virtual/logical topology or a combination of both. The first task of the algorithm is to search for suitable combinations of CPU and memory bricks that can be paired to build EPS virtual/logical topologies. When a VM network request is received in a DC, the first step generates $N_{R_i}^{CPU}$ a set of CPU bricks, $N_{R_i}^{MEM}$ a set of memory bricks and B a VM network request matrix. Next, the set P is used to build sets of possible combinations of CPU bricks. For each subset of P, a set of subsets C is calculated by $\binom{N_{R_i}^{CPU}}{p_{i,1}}$ where each subset C_i contains a possible combination of $p_{i,1}$-CPU bricks. If the generated set C is not empty, for each subset C_i starting from the first, the algorithm searches for a set of memory bricks CM where each of the memory bricks requires a network link from all the CPU bricks in C_i. If the cardinality of CM is greater than or equal to $p_{i,2}$, M which is a set of subsets where each subset M_i contains a possible combination of memory bricks from CM is calculated by $\binom{CM}{p_{i,2}}$. In the event that C is empty, i.e., no CPU brick combination has been generated or the cardinality of CM is less than $p_{i,2}$, the CPU brick number $p_{i,1}$ is discarded and next CPU brick number from the next subset of P is selected and used to generate another set of CPU brick combinations. Figure 6 illustrates an example of generated combinations of CPU and memory bricks with the different subsets of P in Fig. 5, and from a VM network request with 4 CPU bricks {1,3,5,7} and 4 memory bricks {2,4,6,8} to be interconnected.

Afterwards, for each subset in M starting from the first one, if M_i is not empty, the algorithm calculates $tb_{c_{i,j}}$ which is the sum of required bandwidth for network links from a CPU brick in C_i to all memory bricks in M_i. Then, if $tb_{c_{i,j}}$ from a CPU brick in C_i is less than or equal to the capacity of a transceiver $TXcap$, $x_{c_{i,j}}$ is marked as 1, otherwise marked as 0. This step is carried out for all the CPU bricks in C_i. If the summation of the variable $x_{c_{i,j}}$ (for all CPU bricks in C_i) is equal to $p_{i,1}$, the first condition to build an EPS virtual/logical topology is satisfied. After the first condition has been satisfied, the next line calculates $tb_{m_{i,j}}$ which is the sum of required bandwidth of network links from all CPU brick in C_i to a single memory brick in M_i. If $tb_{m_{i,j}}$ is less than or equal to $TXcap$, $x_{m_{i,j}}$ is marked as 1, otherwise marked as 0. Then, if the summation of the variable $x_{m_{i,j}}$ (for all memory bricks in M_i) is equal to $p_{i,2}$, the second condition to build an EPS virtual/logical topology is satisfied. In the event that the summation of the variable $x_{c_{i,j}}$ (for all CPU bricks in C_i) is not equal to $p_{i,1}$ or the summation of the variable $x_{m_{i,j}}$ (for all memory bricks in M_i) is not equal to $p_{i,2}$, the subset of memory bricks is discarded and the next memory brick combination which is the next subset in M selected and checked.

Once the two conditions to build an EPS virtual/logical topology is satisfied, a set V which is the union of CPU bricks in C_i and memory bricks in M_i is generated. The next task of the algorithm is to run the EPS placement strategy algorithm for the selected combinations of CPU bricks and memory bricks in order to select the most suitable bricks or electronic packet switches to perform statistical multiplexing functions when creating an EPS virtual/logical topology. There are three different algorithms that can be selected at this stage: the congestion aware placement strategy algorithm, the

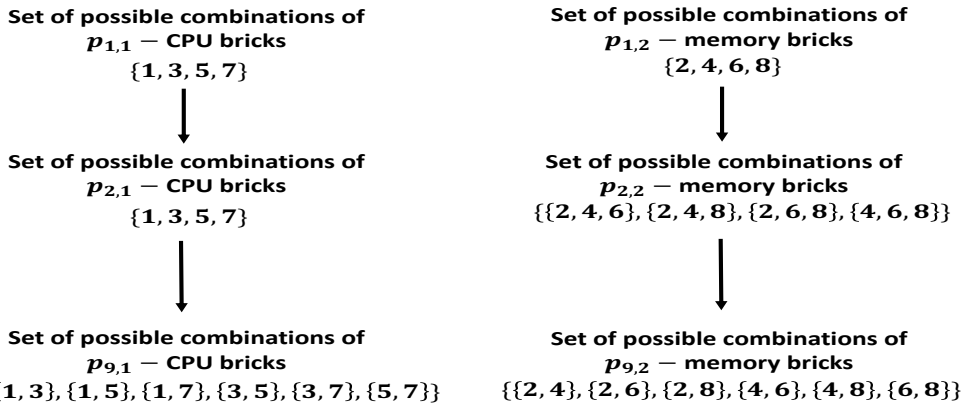

Fig. 6. Example of generated combinations of CPU and memory bricks with the different subsets of P.

Algorithm 1

Inputs: $G = (N, E), R, P$
for each VM network request in R
  Generate $N_{R_i}^{CPU}$, $N_{R_i}^{MEM}$, B
  for each subset of P
    Calculate $C = \binom{N_{R_i}^{CPU}}{p_{i,1}}$
    if $C \neq \emptyset$
      for each subset in C
        Find CM
        if the cardinality of $CM \geq p_{i,2}$
          Calculate $M = \binom{CM}{p_{i,2}}$
          for each subset in M
            if $M_i \neq \emptyset$
              for each CPU brick in C_i
                Calculate $tb_{c_{i,j}}$
                if $tb_{c_{i,j}} \leq TXcap$
                  $x_{c_{i,j}} = 1$
                else if
                  $x_{c_{i,j}} = 0$
                end if
              end for
              if $\sum_{c_{i,j} \in C_i} x_{c_{i,j}} = p_{i,1}$
                for each memory brick in M_i
                  Calculate $tb_{m_{i,j}}$
                  if $tb_{m_{i,j}} \leq TXcap$
                    $x_{m_{i,j}} = 1$
                  else if
                    $x_{m_{i,j}} = 0$
                  end if
                end for
                if $\sum_{m_{i,j} \in M_i} x_{m_{i,j}} = p_{i,2}$
                  $V = C_i \cup M_i$
                  Run EPS placement strategy algorithm
                  Return Output: K
                  for each brick or electronic packet switch in K
                    Build EPS virtual/logical topology and store in TDB
                    Update B
                  end for
                end if
              end if
            end if
          end for
        end if
      end for
    end if
  end for
  if TDB is empty
    Store VM network request as an OCS virtual/logical topology in Z
  else if TDB is not empty
    Compute OT
    if $OT = \emptyset$
      Store EPS virtual/logical topologies in Z
    else if $OT \neq \emptyset$
      Combine OT with all created EPS virtual/logical topologies in TDB and store in Z
    end if
  end if
  for $s = 1: NZ$ (for each topology in Z)
    for each link in the selected topology
      $h_{z_{s,d}} = 0$
      Find already established paths
      if established paths exist
        for each established path
          if $b_{z_{s,d}} + eb_{z_{s,d}} \leq TXcap$
            $h_{z_{s,d}} = 1$
            break
          end if
        end for
      else if established path doesn't exist & $h_{z_{s,d}} = 0$
        Find least congested path
        if resources are available on the found path
          $h_{z_{s,d}} = 1$
        end if
      end if
    end for
    if $\sum_{z_{s,d} \in Z_s} h_{z_{s,d}} = f_{Z_s}$
      Accept and implement VM network request
    else if $\sum_{z_{s,d} \in Z_s} h_{z_{s,d}} \neq f_{Z_s}$ & $s < NZ$
      Check next topology
    else if $\sum_{z_{s,d} \in Z_s} h_{z_{s,d}} \neq f_{Z_s}$ & $s = NZ$
      VM network request is blocked
    end if
  end for
end for

network-hop aware placement strategy algorithm and random placement strategy algorithm. Figure 7(a), 7(b) and 7(c) presents a cluster of dRedBox, 1-Tier-H and 3-Tier-H architectures respectively. Each architecture consists of 2 racks which contain two trays each and each tray contains a CPU and memory brick. In particular, Fig. 7 illustrates examples of different possible EPS virtual/logical topologies that are created from a CPU brick combination of {1,3} and a memory brick combination of {4,8}.

For the dRedBox architecture, in the congestion aware placement strategy algorithm, the brick with the least number of utilized ports is given highest priority to be configured to perform statistical multiplexing functions in the created EPS virtual/logical topology. Algorithm 2 illustrates steps of the congestion aware strategy algorithm for the dRedBox architecture. For each brick in V, the number

Algorithm 2: Congestion aware placement strategy algorithm for dRedBox architecture

for each brick in V
  Calculate the number of utilized brick ports
  Calculate the number of network hops required when the brick is configured as an electronic packet switch
end for
Sort bricks from the brick with least utilized ports to brick with most utilized ports
Find any bricks with the same number of utilized ports
if there are any bricks with same number of utilized ports
  Sort the selected bricks from the brick with the least number of network hops to the brick with most number of network hops
end if
Store sorted bricks in K

of utilized brick ports is calculated. Also, the number of network hops required for each of the bricks to perform statistical multiplexing of flows to and from other bricks is calculated. Next, the bricks are sorted from the brick with the least utilized ports to the brick with the most utilized ports. In a situation where two or more bricks have the same number of utilized ports, those bricks are sorted from the brick which requires the least number of network hops to the brick which requires the most number of network hops. The sorted bricks are stored in K in an order that starts from the most suitable brick to least suitable brick. The network-hop aware placement strategy algorithm follows a similar procedure to the congestion aware placement strategy algorithm. The priority factor for creating an order of suitable bricks to perform statistical multiplexing is the number of required network hops. Thus, after the number of utilized brick ports and required number of network hop for each brick has been calculated, the bricks are sorted from the brick which requires the least number of network hops to the brick which requires the most number of network hops. In a scenario where two or more bricks have the same number of network hops, those bricks are sorted from the brick with the least utilized ports to the bricks with the most utilized ports. In the random placement strategy algorithm, the bricks are selected and sorted randomly without considering any factor. Figure 7(a) shows examples of possible EPS virtual/logical topologies which are created for a CPU brick combination of {1,3} and a memory brick combination of {4,8}. In one of the topologies, a CPU brick is selected to perform statistical multiplexing and while in the other topology a memory brick is selected to perform statistical multiplexing.

For the 1-Tier-H architecture, the statistical multiplexing is performed on the EoT electronic packet switches and not on the bricks. The candidate EoT electronic packet switches that are considered to perform statistical multiplexing are the EoTs switches which are attached to a tray where either a CPU or memory brick in V is located. For instance, in Fig. 7(b), the candidates EoT switches which will be considered to perform statistical multiplexing when building an ESP

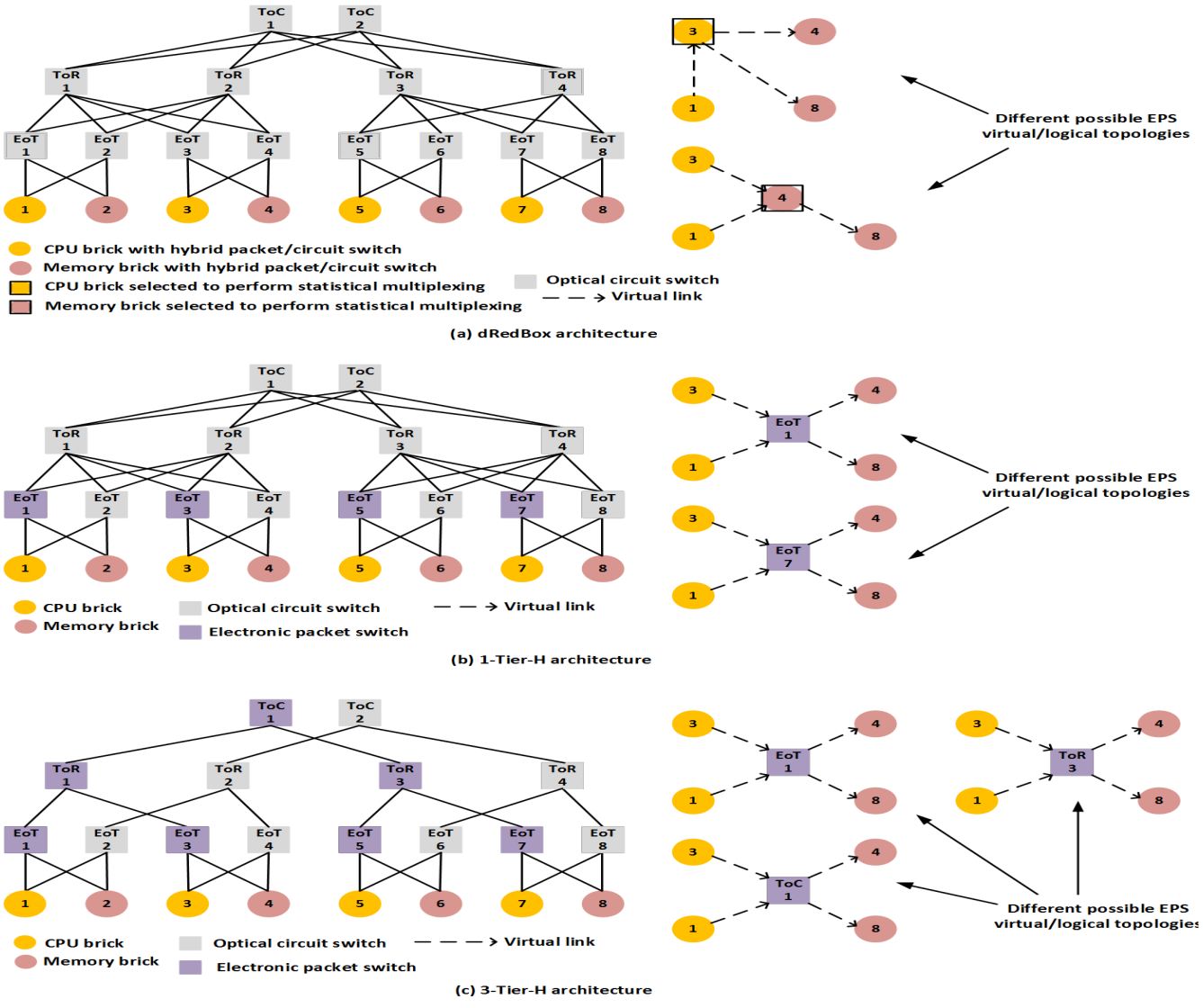

Fig. 7. Examples of different EPS virtual/logical topologies.

virtual/logical topology for a CPU brick combination of {1, 3} and a memory brick combination of {4, 8} are EoT-1, EoT-3 and EoT-7. In the congestion aware algorithm, the number of utilized ports for each of the candidate EoT switches is calculated and the number of network hops required when each of the candidate EoT switches is selected to perform statistical multiplexing of flows to and from the selected CPU and memory bricks is calculated. Next, the switches are sorted from the switch with the least utilized ports to switch with most utilized ports. In a situation where two or more switches have the same number of utilized ports, those switches are sorted from the switch which requires the least number of network hops to the switch which requires the most number of network hops. Next, the sorted EoT switches are stored in K in an order that starts from the most suitable switch to least suitable switch. In the network-hop aware placement strategy algorithm, the candidate EoT electronic switches are sorted from the switch with the least required number of network hop to the switch which requires the most number of network hops. Also, in a situation where two or more EoT electronic packet switches have the same required number of network hops, those set of EoT switches are sorted from the EoT switch with the least utilized ports to the EoT switch with the most utilized ports. In the random placement strategy algorithm, the EoT switches are selected and sorted randomly without considering any factor.

For the 3-Tier-H architecture, statistical multiplexing can occur in EoT, ToR and ToC electronic packet switches. The candidates electronic packet switches that are considered to perform statistical multiplexing are the EoT switches which are attached to a tray where either a CPU or memory brick in V is located, the ToR switches which are attached to a rack where either a CPU or memory brick in V is located, and the ToC electronic packet switch of the cluster where either a CPU or memory brick in V is located. For instance, in Fig. 7(c), the candidates electronic packet switches which will be considered to perform statistical multiplexing when building an ESP virtual/logical topology for a CPU brick combination of {1,3} and a memory brick combination of {4,8} are EoT-1, EoT-3, EoT-7, ToR-1, ToR-3 and ToC-1. The working principle of the congestion aware, network-hop aware and random placement strategy algorithm follows a similar procedure to the 1-Tier-H architecture, the only difference is that since electronic packet switches considered are on different networking layers and have different number of ports, the metric used for determining congestion is the percentage of utilized ports of each of the candidate switches instead of the number of utilized ports.

For all the architectures, after the set K has been created, up to L number of EPS virtual/logical topologies can be created. Next, the EPS virtual/logical topologies that have been built are stored in TDB and the VM network request matrix B is updated by removing all CPU to memory network links that have been used in the created EPS virtual/logical topology. Also, any subset in M that contains any memory brick that has been utilized in the previously built EPS virtual/logical topology with the selected the CPU brick combination in C_i is discarded. After all the subset of P have been checked for all possible combinations of CPU and memory bricks to build EPS virtual/logical topologies, in the event that TDB is empty i.e. no EPS virtual/logical topology has been built, an OCS virtual/logical topology for the VM network request is built and is stored in Z. Otherwise, if TDB is not empty, a set OT which contains any remaining links in B that have not be used to build an EPS virtual/logical topology is generated. If OT is not empty, the links in OT are combined with the EPS virtual/logical topologies in TDB to form hybrid EPS/OCS virtual/logical topologies which are stored in Z. In the event that OT is empty, i.e., there are no remaining links in B to be established, all the EPS virtual/logical topologies in TDB are combined and stored in Z.

C. Network Resource Allocation

The next stage of the algorithm searches for network resources for the created topologies. After Z has been created, the algorithm searches for network resources for the first topology stored in Z. For each link to be established in the selected topology, the algorithm first searches to find an already established path with available bandwidth resources to perform optical grooming services. If there is an existing path with available bandwidth resources, i.e., $b_{z_{s,d}} + eb_{z_{s,d}} \leq TXcap$, the link is marked for optical grooming on the established path ($h_{z_{s,d}} = 1$). Alternatively, if there are existing paths but with no available bandwidth for optical grooming or there are no existing paths for optical grooming, the algorithm searches for free resources (I/O and network ports) by finding and selecting the least congested network path. If resources are available on the found network path, the link is marked to be deployed on the resources on that path ($h_{z_{s,d}} = 1$). If resources are available for all links, the VM network request is accepted. If resources are not available for any link, other stored topologies in Z are checked. If resources are not available to any of the topologies, the VM network request is blocked.

VI. Simulation and Results

In this section, a performance analysis of the dRedBox, 1-Tier-H and 3-Tier-H architectures are presented and discussed. The configuration of the dRedBox, 1-Tier-H and 3-Tier-H DC architecture is simulated as illustrated in Fig. 1. Each of the DC architectures consists of a cluster containing 4 racks with each rack consisting of 4 heterogeneous trays. Each tray is populated with 14 bricks, i.e., 7 CPU and 7 memory bricks. Each brick is interfaced with 24 10Gb/s transceivers. All the DC architectures have a 1:1 port connectivity subscription ratio between all networking layers and the same number of switch ports on all networking layers. Each of the EoT switches has 672 ports (i.e. the same performance of each EoT switch simulated can be achieved by connecting 8 switches with 84 ports each in parallel), each of the ToR switches has 2688 ports (i.e. the same performance of each ToR switch simulated can be achieved by connecting 12 switches with 224 ports each in parallel) and each of the ToC switches has 5376 ports (i.e. the same performance of each ToR switch simulated can be achieved by connecting 16 switches with 336 ports each in parallel).

There are different bandwidth requirements for local memory associated with the memory technology and processor technology. As mentioned in section II, results from a previous study and experiment in [15] showed that for certain applications, 20-40 Gb/s for remote memory access can achieve minimal (under 10%) application performance degradation. Furthermore, a related study in [16] reported that a bandwidth of 582 Mib/s (~ 5Gb/s) can be realized from a single CPU core to remote access memory. Also, the authors in [17] demonstrated remote memory access with 10 Gb/s links and up to 68% sustained memory bandwidth. In this study, VM network requests are randomly generated using the following parameters: 2 to 5 number of CPU bricks, 2 to 5 number of memory bricks and bandwidth requirement between a CPU brick and a memory brick that varies between 1 to 5 Gb/s and 6 to 10 Gb/s which is classified as low bandwidth traffic pattern (LBTP) and high bandwidth traffic pattern (HBTP) respectively (i.e. for a complete round trip transaction between one CPU and one memory brick, LBTP is 2 to 10 Gb/s and HBTP is 12 to 20 Gb/s). The authors in [7] used a similar bandwidth traffic pattern to evaluate the benefits of an OPS/OCS hybrid DC. Also, for each generated VM network request, each of the CPU brick requires a network link to each of the memory bricks. We assume VM network requests arrive dynamically following a Poison process with a mean inter-arrival rate of 10 time units. A previous study on disaggregated DCs in [13] has used a similar approach. A holding time range of 525 to 2100 time units with increments of 525 time units was selected to ensure different levels of network load in the DC. Once the holding time of a successfully deployed VM network request expires, the resource attached to that VM network request are released.

A total of 400 VM network requests are generated and the results obtained in this paper are averaging 4 simulations runs for each point with a 95% confidence interval. The following abbreviations are used for the different EPS function placement strategies on the various architectures. dRedBox congestion aware, dRedBox network-hop aware and dRedBox random placement strategies are represented as D-COS, D-NES and D-RAS respectively. The 1-Tier-H congestion aware, 1-Tier-H network-hop aware and 1-Tier-H random placement strategies are represented as 1-T-COS, 1-T-NES and 1-T-RAS respectively. Finally, 3-Tier-H congestion aware, 3-Tier-H network hop aware and 3-Tier-H random placement strategies are represented as 3-T-COS, 3-T-NES and 3-T-RAS respectively. The various EPS placement strategies are evaluated in terms of blocking probability, network capacity, network utilization, energy efficiency and cost. Blocking probability is the ratio of number of blocked requests to the total number of requests that have been processed in the DC network. Network capacity is a measure of the total number of bits per second (b/s) transmitted on all the network links of DC network. Network utilization is the ratio of the number of utilized network resources (number of used brick I/O ports and switch ports) to the total number of network resources (total number of brick I/O ports and switch ports in the DC network). The parameters describing the cost and power consumption of components are set according to the values presented in Table II, it assumed that the cost of a 10G transceiver is $50, and the power per optical switch port is 0.05W [22], the remaining values are sourced from [4, 9]. Also, the cost and power consumption are calculated based on the number of resources used.

TABLE II

Component	Power (W)	Cost ($)
Optical switch port	0.05	500
Electronic switch port	12.5	500
Transceiver 10G	1	50

A. *Random Bandwidth Traffic Variation*

For the random bandwidth traffic variation simulation scenario, the VM network requests generated are randomly assigned to either LBTP or HBTP. Figure 8 shows the blocking probability of the different EPS placement strategies across the dRedBox, 1-Tier-H and 3-Tier-H architectures at different holding times. The points plotted in Fig. 8 represent the blocking probability after the last VM network request, i.e., 400th request, has been received and processed in the DC network. It can be noted from Fig. 8 that the dRedBox architecture for all the EPS placement strategies has a lower blocking probability than the 1-Tier-H and 3-Tier-H architectures. Furthermore, despite the 3-Tier-H architecture having more levels for EPS than the 1-Tier-H architecture, Fig. 8 shows that the 3-Tier-H and 1-Tier-H architecture display similar levels of blocking probability across all EPS placement strategies and holding times. This is because both the 3-Tier-H and 1-Tier-H architecture have the same proportions of brick I/O ports dedicated to either optical circuit or electronic packet network, this limits resource availability to process different traffic variations (LBTH or HBTH). Thus, having different levels of EPS

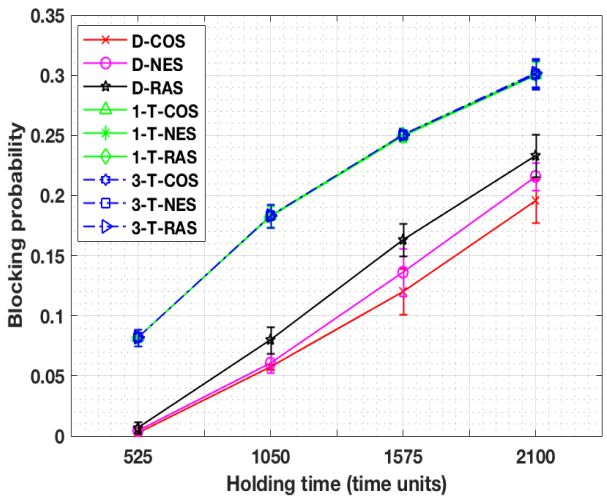

Fig. 8. Blocking probability for random bandwidth traffic variation.

switches in the 3-Tier-H architecture has negligible impact on resource availability to serve VMs. In addition, it can be observed that D-COS performs the best (has the lowest blocking probability). At 2100 holding time units, D-COS demonstrates approximately 9% lower blocking probability than D-NES, 16% lower blocking probability than D-RAS and 35% lower blocking probability than the 1-T-COS, 1-T-NES, 3-T-RAS, 3-T-COS, 3-T-NES and 3-T-RAS.

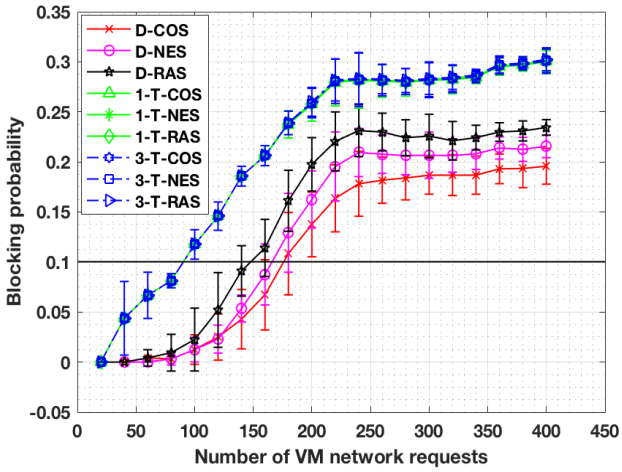

Fig. 9. Blocking probability for 2100 holding time scenario.

Figure 9 shows the blocking probability of the 2100 holding time scenario with a line highlighting the 10% (0.1) blocking probability point. It can be observed from Fig. 9, that the dRedBox architecture for all the EPS placement strategies demonstrates lower blocking probability than the 1-Tier-H and 3-Tier-H architectures across different number of VM network requests. The 0.1 blocking probability point is selected as the maximum threshold value for blocking probability to evaluate the performance of the network.

Figure 10(a)-(d) displays the number of successfully deployed VM network requests, network capacity, network utilization and energy efficiency respectively, from the 2100 holding time unit (Fig. 9) scenario at 10% blocking probability. Numerous insights can be noted from Fig. 10. For the dRedBox architecture, in terms of the number of successfully deployed VM network requests, D-COS has deployed approximately 5% more than D-NES and 16 % more than D-RAS. A similar trend is observed in terms of network capacity and network utilization. Furthermore, D-COS, D-NES and D-RAS demonstrate approximately the same performance in terms of energy efficiency. This implies that D-COS demonstrates the overall best performance, because at 10% blocking probability, D-COS has the highest number of successfully deployed VM network requests while

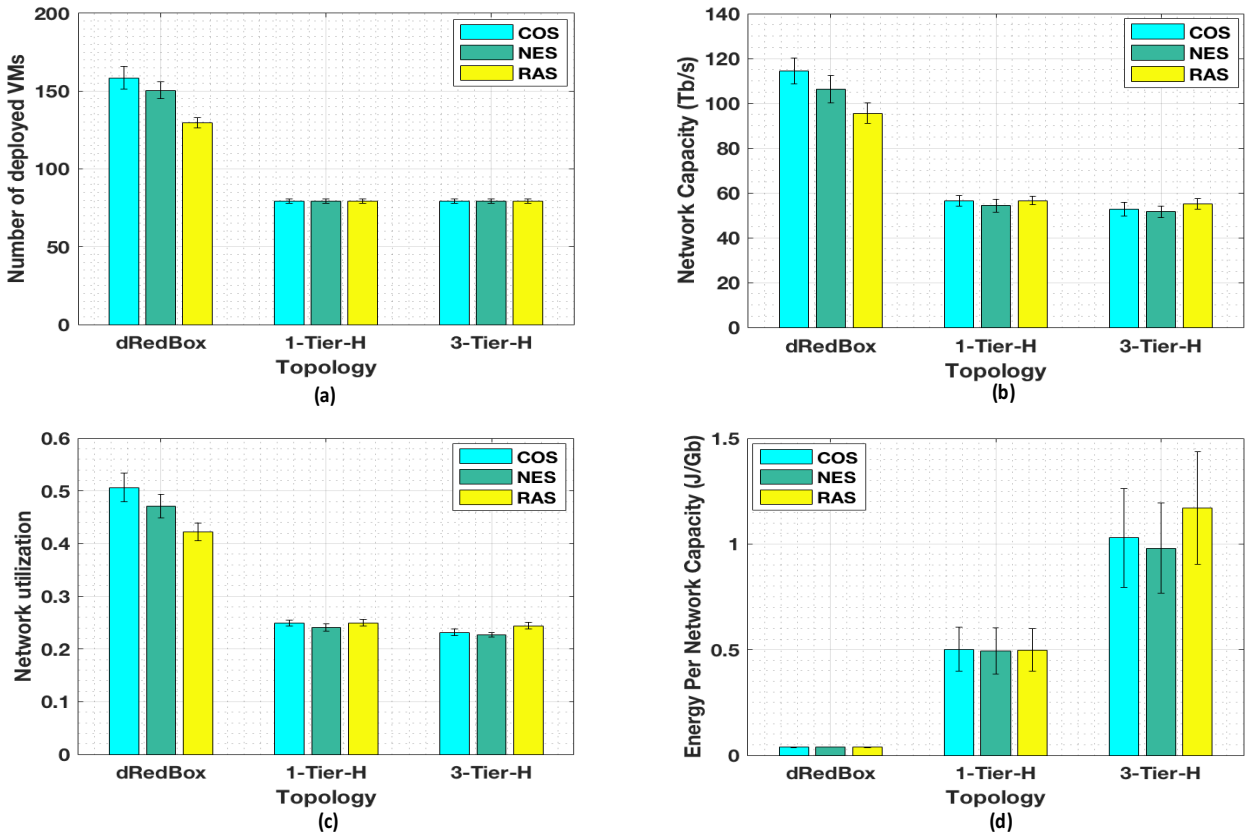

Fig. 10. Performance indicators at 10% blocking probability for dRedBox, 1-Tier-H and 3-Tier-H architectures: (a) Number of successfully deployed VM network requests (b)Network capacity, (c) Network utilization, (d) Energy per network capacity.

achieving approximately the same energy efficiency with D-NES and D-RAS. For the 1-Tier-H, it is noted from Fig. 10 that 1-T-NES has the same number of successfully deployed VM network requests with 1-T-COS and 1-T-RAS but at a slightly lower network capacity and network utilization. Also for the 3-Tier-H, it is noted from Fig. 10 that 3-T-NES has the same number of successfully deployed VM network requests with 1-T-COS and 1-T-RAS but at a slightly lower network capacity and network utilization. Therefore, for the 1-Tier-H and 3-Tier-H architectures, the network-hop aware placement strategy uses resources more efficiently than the congestion aware and random placement strategy to achieve the same level of blocking probability. This translates into slightly better energy efficiency improvements as depicted in Fig. 10(d). 1-T-NES demonstrates about 2% energy savings in comparison to 1-T-COS and 1% energy savings in comparison to 1-T-RAS, while 3-T-NES demonstrates about 5% energy savings in comparison to 3-T-COS and 16% in comparison to 3-T-RAS. Comparing the performance of the dRedBox, 1-Tier-H and 3-Tier-H architectures, in terms of the number of successfully deployed VM network requests (Fig. 10(a)), the dRedBox architecture demonstrates a better performance than the 1-Tier-H and 3-Tier-H architectures for all EPS placement strategies. In particular, the D-COS has implemented about 100% more VM network requests than 1-T-COS, 1-T-NES, 1-T-RAS, 3-T-COS, 3-T-NES and 3-T-RAS. Furthermore, in terms of energy efficiency, dRedBox architecture performs the best, the 1-Tier-H architecture ranks second and the 3-Tier-H architecture has the worst performance for all EPS placement strategies. D-COS has energy savings of approximately 92% in comparison to 1-T-NES and 96% in comparison to 3-T-NES. This is because dRedBox is supported by a pure optical network while 1-Tier-H consist of power hungry electronic switches at the tray level and 3-Tier-H consist of power hungry electronic switches at the tray, rack and cluster level. It can also be noted from Fig. 10(c) that at 10% blocking probability, the 3-T-NES and 3-T-COS achieves the same number of successfully deployed VMs with the 1-T-NES but at lower network utilization. In particular, 3-T-NES demonstrates about 6% lower network utilization in comparison to 1-T-NES while 3-T-COS demonstrates about 4% lower network utilization in comparison to 1-T-NES. The lower network utilization demonstrated by the 3-T-NES and 3-T-COS compared to 1-T-NES is because the 3-Tier-H architecture can perform EPS on the tray, rack and cluster networking level while the 1-Tier-H architecture can only perform EPS on the tray level. Thus, 3-Tier-H requires less network hops to implement EPS virtual/logical topologies than the 1-Tier-H. However, this comes at a demerit of increased power consumption and a decrease in energy efficiency (see Fig. 10(d)) because of multiple layers of power hungry electronic switches.

B. Proportional Bandwidth Traffic Variation

For the second simulation scenario, 2100 holding time units is used and different percentage ratios of the total generated VM network requests are assigned between LTBP and HTBP. In addition, all the points plotted in the figures presented in this section represent the points after the last VM network request, i.e., 400th request, has been received and processed in the DC network. Figure 11 displays the blocking probability for all architectures across different bandwidth percentage ratios. It can be observed from Fig. 11 that D-COS, D-NES and D-RAS demonstrates lower blocking probability than the 1-Tier-H and 3-Tier-H architectures from 50:50 percentage ratio to 0:100 percentage ratio, while 1-T-COS, 1-T-NES, 1-T-RAS, 3-T-COS, 3-T-NES and 3-T-RAS demonstrates similar blocking probability across all bandwidth percentage ratios. In more detail, the D-COS demonstrates the lowest blocking probability for all bandwidth percentage ratios except at 0:100 where it is equal to D-NES and D-RAS. This is because at 0:100 bandwidth percentage ratio, all VM network requests are served by only OCS virtual/logical topologies (i.e. the EPS placement strategies have no effect because no EPS virtual/logical topology is built). Additionally, the more increase in the % share of LBTP, the lower the blocking probability, this is because as the percentage share of LBTP increases, the greater the probability of forming EPS virtual/logical

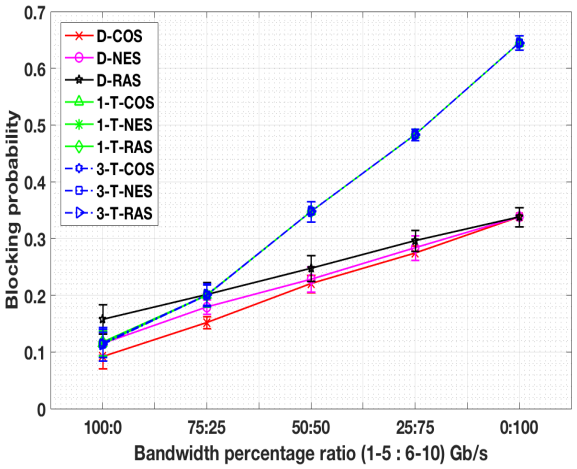

Fig. 11. Blocking probability for proportional bandwidth traffic variation.

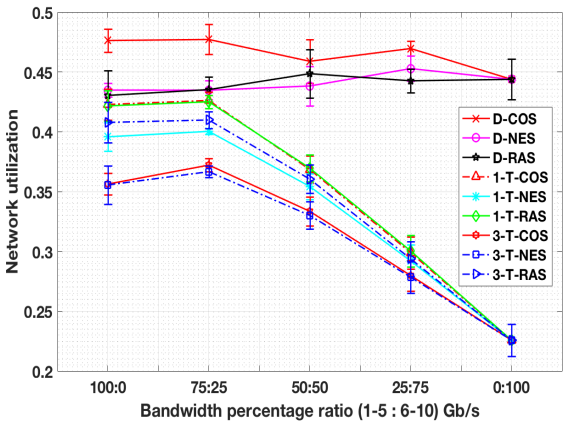

Fig. 12. Network utilization for proportional bandwidth traffic variation.

topologies, which in turns leads to conservation of more resources to accommodate future request.

Figure 12 shows the network utilization across different bandwidth percentage ratios. It is noted that the network utilization of the D-COS, D-NES and D-RAS remains approximately constant. On the other hand, the network utilization for 1-T-COS, 1-T-NES, 1-T-RAS, 3-T-COS, 3-T-NES and 3-T-RAS decreases as the percentage share of LBTP decreases from 75% to 0%. The approximately constant utilization demonstrated by dRedBox architecture is as a result of its ability to re-configure and deploy either packet/circuit switching services to any I/O ports to handle traffic variations. Therefore, as the bandwidth percentage ratio between LBTP to HBTP varies, the ratio between packet to circuit switch port utilization on the bricks also varies to match the traffic variations. While the drop of network utilization demonstrated by the 1-Tier-H and 3-Tier-H architecture is due to the fact that the I/O ports are fixed and dedicated to either OCS or EPS. This confirms the high blocking probability demonstrated by the 1-Tier-H and 3-Tier-H architecture in Fig 11.

Comparing the 1-Tier-H and 3-Tier-H architecture, it can be observed from Fig. 12 that 3-T-COS and 3-T-NES demonstrates a lower network utilization in comparison to the 1-T-NES from 100:0 to 25:75 bandwidth percentage ratio, this because the 3-Tier-H architecture can perform EPS on more networking levels (tray, rack and cluster) while the 1-Tier-H architecture can perform EPS at only the tray level. However, the benefits of more layers of EPS switches in the 3-Tier-H architecture comes at a cost of increased power consumption and a decrease in energy efficiency when compared to the 1-Tier-H architecture which is supported by a pure optical network on the rack and cluster network layer. This evaluation confirms the trend demonstrated in Fig.10 and discussed in the random bandwidth variation scenario regarding the trade-off between energy and network utilization between the 1-Tier-H and 3-Tier-H architecture. Thus, the following conclusions can be made for the 1-Tier-H and 3-Tier-H architectures: First, for each of the architectures, the network-hop aware strategy uses less networking resources that than the congestion and random aware placement strategies while delivering similar levels of blocking probability. Secondly, at the same level of blocking probability, the 3-Tier-H architecture uses less networking resources than the 1-Tier-H architecture because of the advantage of having more network layers to perform EPS, however this comes at a cost of increased power consumption which leads to a decrease in energy efficiency.

C. *Cost Analysis*

Figure 13 displays the cost of the number of transceivers in dRedBox, 1-Tier-H and 3-Tier-H architectures across different number of racks. To evaluate the difference in cost of the different DC architectures, only the cost of transceivers is considered for two reasons. Firstly, there is no extra cost in embedding the hybrid and programmable packet/circuit switch on each brick because it is fabricated as part of the MPSoC hardware of the CPU brick where the CPUs resides or the memory brick where the memory controller resides. Instead, software is deployed on the programmable logic of the MPSoC to provide packet/circuit services [17]. Secondly, the price of an optical switch port and electronic switch port are considered equal based on the references provided (see Table II) and the total number of network ports for each of the various architectures are equal. For the dRedBox architecture, the total cost is equal to the cost of all the transceivers embedded on the bricks. For the 1-Tier-H architecture, the total cost is equal to the cost of all the transceivers embedded on the bricks and EoT electronic packet switches. For the 3-Tier-H architecture, the total cost is equal to the cost of all the transceivers embedded on the bricks, EoT electronic packet switches, ToR electronic packet switches and ToC electronic packet switches. The results clearly show that dRedBox is the least costly architecture, followed by 1-Tier-H architecture and 3-Tier-H architecture which is the most expensive. In more detail, the dRedBox has a 50% cost reduction in comparison to 1-Tier-H and an 80% cost reduction in comparison to 3-Tier-H at 4 racks. The improvement in cost savings for the dRedBox architecture in comparison to the 1-Tier-H and 3-Tier-H architecture can also be observed for 8 and 12 number of racks. The reason for the significant reduction in the cost of the dRedBox is due to the additional transceivers present on the electronic switches at the tray level in 1-Tier-H and at the tray, rack and cluster level in 3-Tier-H.

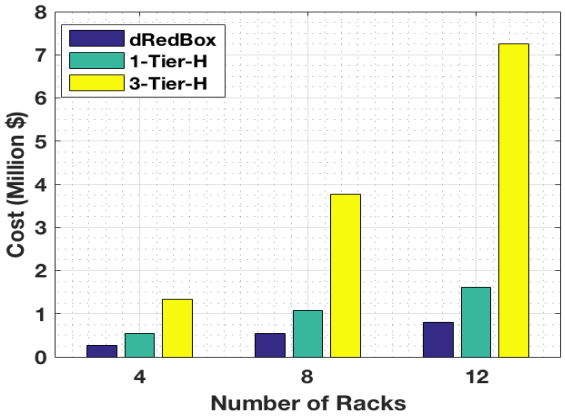

Fig. 13. Cost of transceivers.

VII. Conclusion

In this paper, algorithms and networking strategies were proposed to build custom network topologies and deploy network service chains of EPS/OCS function services across various disaggregated DC network to serve VM network requests. The performance of the various EPS placement strategies across the dRedBox, 1-Tier-H and 3-Tier-H architectures were evaluated under different traffic pattern scenarios. Extensive analysis and results show that for the dRedBox architecture, the congestion aware placement strategy demonstrates the best performance in terms of blocking probability and number of successfully deployed VM requests at 10% blocking probability. Furthermore, for the 1-Tier-H and the 3-Tier-H architectures, the network hop

placement strategy performs best because it uses less network capacity and networking resources than the congestion aware and random placement strategies, but demonstrates a similar performance in blocking probability. This translates to energy efficiency savings when compared to the congestion aware and random placement strategies. Furthermore, comparing the dRedBox, 1-Tier-H and 3-Tier-H architectures, the dRedBox architecture delivers substantially better performance than the 1-Tier-H and 3-Tier-H architectures. In terms of blocking probability, the dRedBox architecture demonstrated about 35% decrease in blocking probability compared to the 1-Tier-H and the 3-Tier-H architectures. In terms of energy efficiency, the dRedBox architecture demonstrated 92% energy savings in comparison to the 1-Tier-H architecture and 96% in comparison to the 3-Tier-H architecture. Finally, in terms of cost, the dRedBox is the least expensive architecture with about 50% and 80% cost savings in comparison to the 1-Tier-H and 3-Tier-H architectures respectively. In conclusion, the demonstrated benefits of the dRedBox architecture and insightful gains into the performance of the various networking strategies have provided significant information and solutions to overcoming: the limitations of conventional hybrid architectures and the challenges for networking designs of disaggregated DCs.

ACKNOWLEDGMENT

The work was partially supported by European Union's H2020 funded dRedBox project with grant agreement No.687632., as well as EPSRC projects EP/L026155/2 INSIGHT and EP/R035342/1 TRANSNET.

REFERENCES

[1] Cisco Global Cloud Index: Forecast and Methodology, 2015–2020.

[2] S. Han et al., "Network support for resource disaggregation in next-generation datacenters," *in 12th ACM Workshop on Hot Topics in Networks HotNets*, 2013.

[3] K. Katrinis, et al., "Rack-scale Disaggregated cloud data centers: The dReDBox project vision", *in 2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2016, pp. 690–695.

[4] N. Farrington, et al., Helios: a hybrid electronic/optical switch architecture for modular data centers, *ACM SIGCOMM Computer. Commun*. Rev. 41 (4) (2011), 339–350.

[5] G. Wang, D.G. Andersen, M. Kaminsky, K. Papagiannaki, T.S. Ng, M. Kozuch, M. Ryan, c-Through: part-time optics in data centers, *in ACM SIGCOMM Computer Communication Review*, vol. 40, 2010.

[6] K. Christodoulopoulos, D. Lugones, K. Katrinis, M. Ruffini and D. O'Mahony, "Performance evaluation of a hybrid optical/electrical interconnect," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 3, pp. 193-204, March 2015.

[7] A. Pages, M. P. Sanchis, S. Peng, J. Perello, D. Simeonidou and S. Spadaro, "Optimal virtual slice composition toward multi-tenancy over hybrid OCS/OPS data center networks," in *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 10, pp. 974-986, 2015.

[8] S. Peng et al., "Multi-Tenant Software-Defined Hybrid Optical Switched Data Centre," in *Journal of Lightwave Technology*, vol. 33, no. 15, pp. 3224-3233, 2015.

[9] M. Imran et al., "Performance evaluation of hybrid optical switch architecture for data center networks," *Opt. Switching Netw*. (2016).

[10] Q. Chen, V. Mishra and G. Zervas, "Reconfigurable computing for network function virtualization: A protocol independent switch," in *International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Cancun, 2016, pp. 1-6.

[11] Albert Pages, Rubén Serrano, Jordi Perelló, and Salvatore Spadaro, "On the benefits of resource disaggregation for virtual data centre provisioning in optical data centres," *Computer Communications*, vol. 107, pp. 60–74, April 2017.

[12] H. M. Mohammad Ali, T. E. H. El-Gorashi, A. Q. Lawey and J. M. H. Elmirghani, "Future Energy Efficient Data Centers With Disaggregated Servers," in *Journal of Lightwave Technology*, vol. 35, no. 24, pp. 5361-5380, 2017.

[13] G. Zervas, H. Yuan, A. Saljoghei, Q. Chen and V. Mishra, "Optically disaggregated data centers with minimal remote memory latency: Technologies, architectures, and resource allocation [Invited]," *in IEEE/OSA Journal of Optical Communications and Networking,* vol. 10, no. 2, pp. A270-A285, Feb. 2018.

[14] A. Peters and G. Zervas, "Network synthesis of a topology reconfigurable disaggregated rack scale datacentre for multi-tenancy," in *Optical Fiber Communications Conference and Exhibition (OFC)*, Los Angeles, CA, 2017.

[15] P. X. Gao et al., "Network Requirements for Resource Disaggregation This paper is included in the Proceedings of the," *in Proc. 12th USENIX Symp. Oper. Syst. Des. Implement*. (OSDI), 2016.

[16] D. Syrivelis, et al., "A Software-defined Architecture and Prototype for Disaggregated Memory Rack Scale Systems", *in International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation,* 2017.

[17] A. Saljoghei, V. Mishra, M. Bielski, I. Syrigos, K. Katrinis, D. Syrivelis, A. Reale, D. N. Pnevmatikatos, D. Theodoropoulos, M. Enrico, N. Parsons, and G. Zervas, "dRedDbox: Demonstrating Disaggregated Memory in an Optical Data Centre," *in Optical Fiber Communication Conference*, 2018, paper W1C.1

[18] Yan Yan, George M. Saridis, Yi Shu, Bijan Rahimzadeh Rofoee, Shuangyi Yan, Murat Arslan, Thomas Bradley, Natalie V. Wheeler, Nicholas Heng-Loong Wong, Francesco Poletti, Marco N. Petrovich, David J. Richardson, Simon Poole, George Zervas, and Dimitra Simeonidou, "All-Optical Programmable Disaggregated Data Centre Network Realized by FPGA-Based Switch and Interface Card," *in J. Lightwave Technol.* 34, 1925-1932 (2016).

[19] G. M. Saridis et al., "EVROS: All-optical programmable disaggregated data centre interconnect utilizing hollow-core bandgap fibre," *in European Conference on Optical Communication (ECOC)*, Valencia, 2015.

[20] C. Kachris and I. Tomkos, "A Survey on Optical Interconnects for Data Centers," *in IEEE Communications Surveys & Tutorials,* vol. 14, no. 4, pp. 1021-1036.

[21] Q. Chen, V. Mishra, N. Parsons and G. Zervas, "Hardware programmable network function service chain on optical rack-scale data centers," *in* Optical Fiber Communications Conference and Exhibition (OFC*)*, Los Angeles, CA, 2017.

[22] Polatis Series 6000 48xCC OSM [Online]. Available at: http://www.polatis.com/switch-modules-for-oem-all-optical-switch-module-solutions-original-equipment-manufactures.asp.